# EDA: Tasks, Tools, Principles

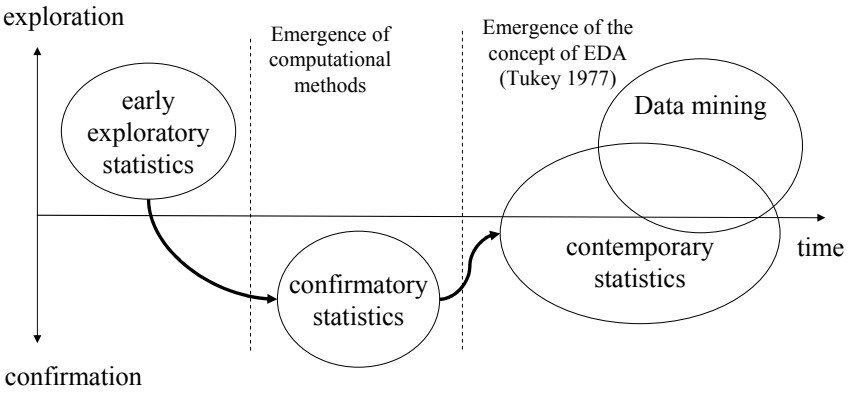
Natalia Andrienko & Gennady Andrienko
Fraunhofer Institute AIS
Sankt Augustin
Germany
http://www.ais.fraunhofer.de/and


*Potsdam, 27.09.2005*

# Presentation Plan

- Introduction
  - What is EDA?
  - Examples of tools for EDA (demo)
  - Our ambitions
- Our theory of EDA
  - General structure of data
  - Tasks
  - Principles
  - Top-down and bottom-up processes in EDA
- Conclusion
  - The theory for a dual use
  - Open issues

# Exploratory Data Analysis (EDA) and Evolution of Statistics

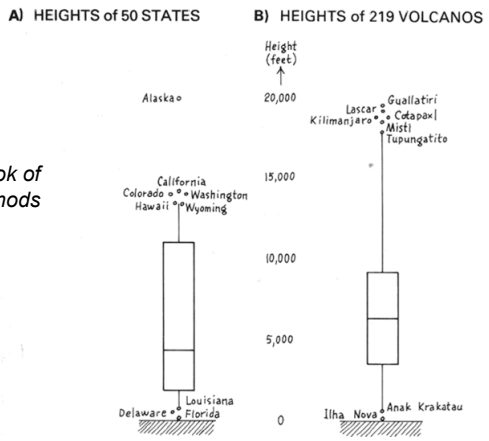



Tukey saw EDA as a return to the original goals of statistics, i.e. *detecting and describing patterns, trends, and relationships in data* and *generation of hypotheses.*

# EDA and Visualization

…by its very nature the main role of EDA is to open-mindedly explore, and graphics gives the analysts unparalleled power to do so…

*NIST/SEMATECH e-Handbook of Statistical Methods*



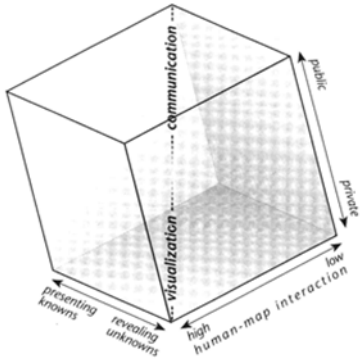

The greatest value of a picture is when it *forces* us to notice what we never expected to see.

*John W. Tukey*

# EDA and Cartographic Visualization

Cartography[3]

International Cartographic Association
Commission on Visualization

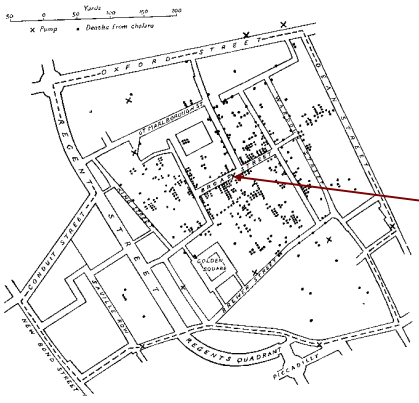



*Alan MacEachren* 1994

…emphasis on the role of **highly interactive maps** in individual and small group efforts **at hypothesis generation, data analysis, and decision-support**.

*A.M.MacEachren* and *M.-J. Kraak* 1997

# An Example of Cartographically-Supported Spatial EDA




Dr. John Snow

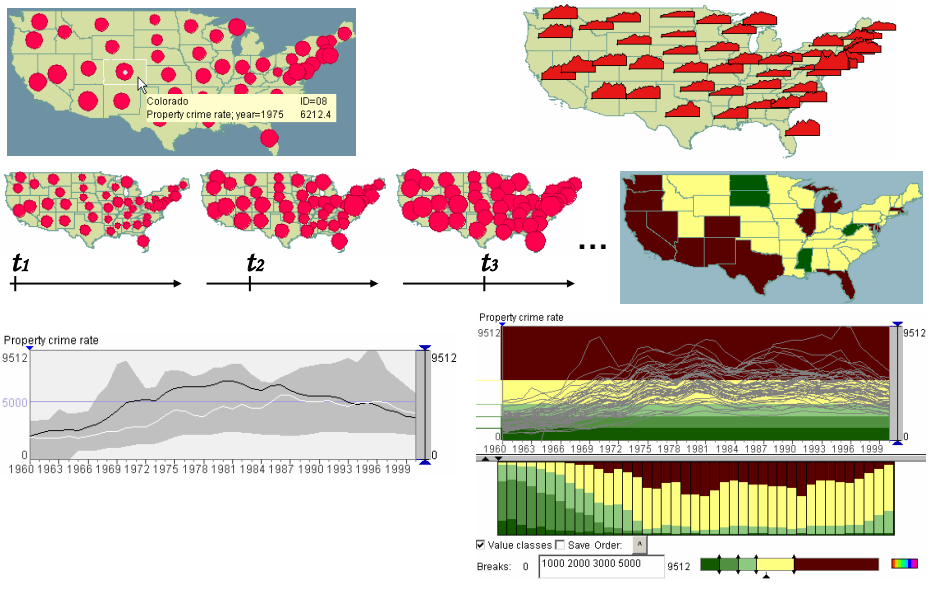
Map of locations of deaths from cholera
London, September 1854

infected water pump?

# Current EDA Tools

- Information visualisation software such as Dynamic Query, TreeMap, and TimeSearcher from HCIL, Univ. Maryland (Ben Shneiderman)
- Geovisualisation tools such as GeoVistaStudio (Penn State Univ.) and Descartes/CommonGIS (Fraunhofer Institute AIS)
- Graphical statistics tools, for example, Manet and Mondrian (Augsburg Univ.)
  - ☞ Usually such systems are research prototypes that implement innovative ideas but provide restricted functionality and limited user support

# Examples of tools for EDA (demo)



# Research Problems

- How do we (tool designers) know what tools are needed? (i.e. what capabilities should be provided)
- What are the best ways to combine several tools providing complementary capabilities?
- How can we teach the users when and how to apply what tools?

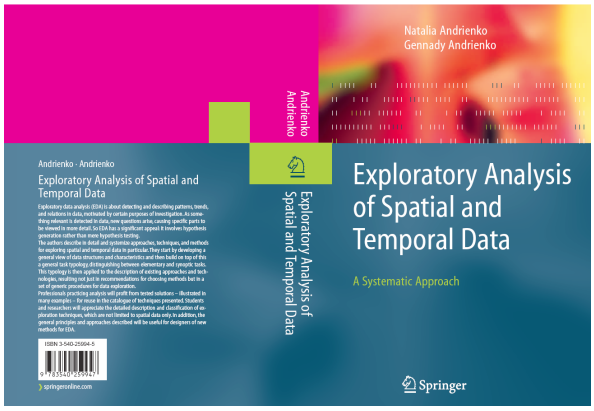
We have a practical experience from many cases of choosing or designing tools to analyse various datasets given to us.

We have also experience in demonstrating users how to analyse their data

And now we want to generalise our experiences and to turn the practice into a theory

# EDA: from Practice to Theory

- ✓Data
- ✓Tasks
- ✓Tools
- ✓Principles




*to appear ≈ end 2005*

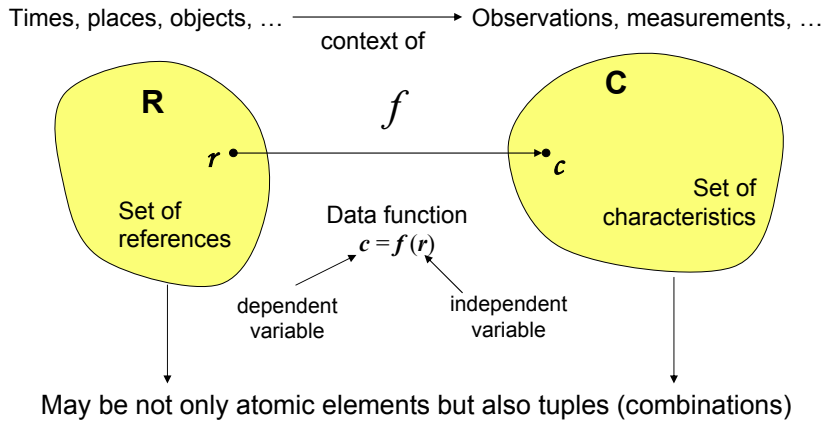
# EDA: Our Theory

- Data
  - A general model of data: $f: \mathrm{R} \rightarrow \mathrm{C}$ (a mapping from references to characteristics)
- Tasks
  - A general model of task: target + constraints
  - Task levels: elementary (individual references and characteristics) and synoptic (sets of references and behaviours of characteristics)
- Tools
  - Tool catalogue: visualisation, display manipulation, data manipulation, querying, computation
  - Modes and mechanisms for tool combination
- Principles
  - To guide tool developers in tool/system design
  - To guide data analysts in choosing and using the tools

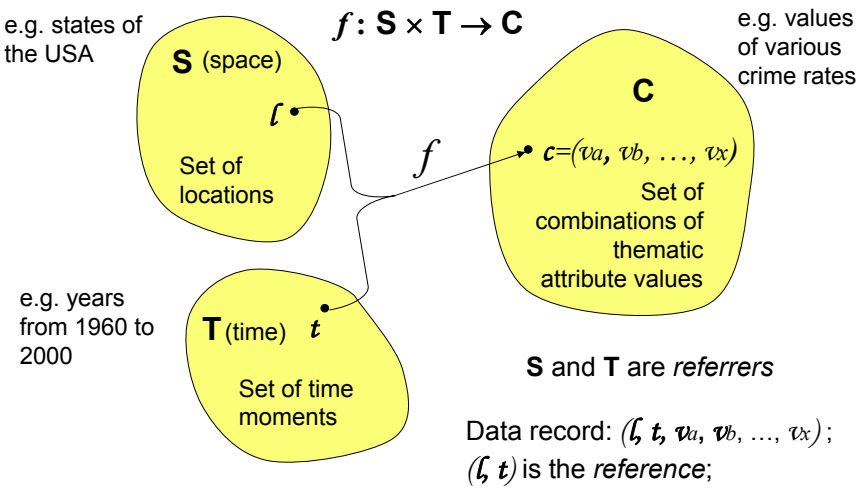
# The Task-Centred Approach

- EDA consists of *tasks*, i.e. finding answers to various *questions* about data.
- To find the answers, an analyst needs appropriate *tools*.
- To create appropriate tools, a designer must know the tasks.
  - The variety of possible tasks typically requires combining several tools.
- An analyst needs understanding what tools to choose for what tasks.
- We want to describe the tasks of EDA in a general and comprehensive way.
  - The tasks serve as a basis for establishing the principles.

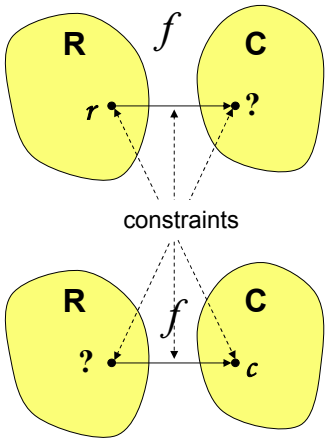
# The General Data Model



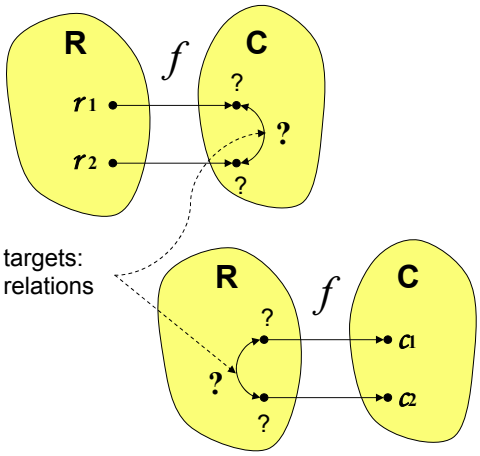

# Two-Dimensional Data (Example)



# Elementary Tasks

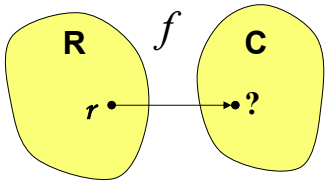



Lookup (direct, inverse)

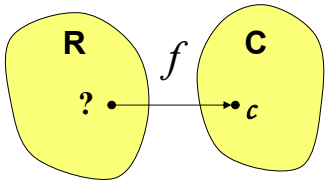



Comparison (direct, inverse)

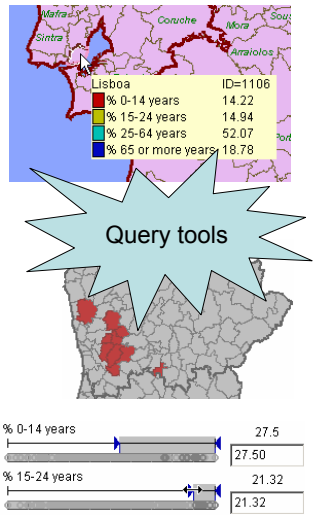
# Support of Lookup Tasks



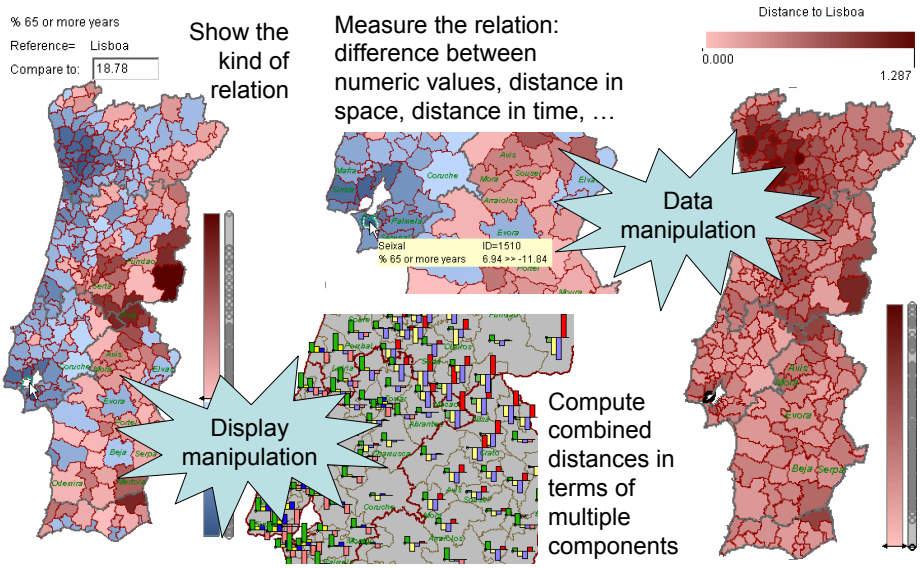

Tool: allows the user to specify or locate $r$; shows or allows the user to determine $c$




Tool: allows the user to specify $c$; shows or allows the user to locate $r$



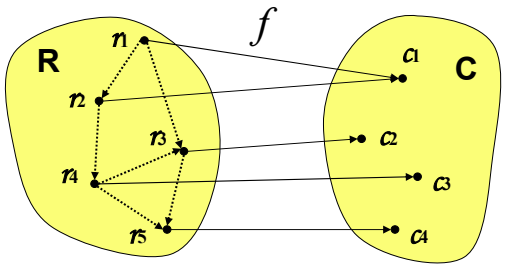
# Support of Comparison Tasks



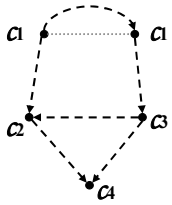

# Elementary Tasks (Summary)

- ✓ Relatively easy to do
- ✓ Well supported by tools: querying, display manipulation (e.g. visual comparison), data manipulation (e.g. computing differences, changes, multi-dimensional distances…)
- − But play only a subordinate role in EDA

# Synoptic Level




References and relations between them are considered all together as a unit

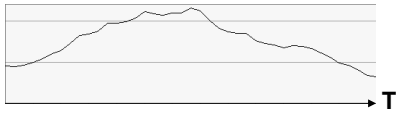



The *behaviour* of $f$ over **R**: the configuration of characteristics corresponding to all references in **R** and the relations between them

Example

$f$: **T** → **N**

**T**: time (linearly ordered set of moments)
**N**: set of numbers,
values of a numeric attribute $f$(t)



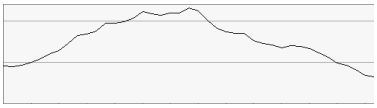

The behaviour of the attribute over **T**

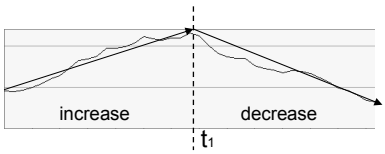
# The Task of Behaviour Characterisation

$f$: **R** → **C**

- Describe the behaviour of the data function (attribute, group of attributes) over the reference set **R** (or subset **R′**).

= Represent the behaviour by an appropriate *pattern*



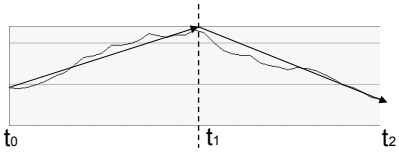



E.g. a verbal pattern: "increase from $x_1$ to $x_2$ over the period from $t_0$ to $t_1$, then decrease to $x_3$ over the period from $t_1$ to $t_2$".
A summary pattern: min, max, mean, …
A formula
A graphical pattern
…

A *compound* pattern; consists of 2 subpatterns

# Other Synoptic Tasks

- Behaviour (pattern) search:
  - find the subset(s) of the reference set where a given behaviour (specified by a pattern) takes place, e.g. find the intervals of value increase
- Behaviour comparison:
  - Determine the kind of (same, different, opposite) and characterise and/or measure the relation between behaviours
    - Of one function (attribute, attribute group) over two or more reference subsets
    - Of two or more functions over the same reference (sub)set
    - Of two or more functions over different reference subsets

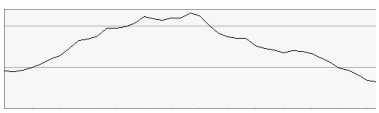



E.g. the behaviour over $[t_1, t_2]$ is opposite to the behaviour over $[t_0, t_1]$ and the change is about 1.5 times faster

# The Primary Task of EDA

- Characterise the behaviour of the data function over the entire reference set

  ⇒The tool to support: 1) allows the user to see the entire reference set and all the corresponding characteristics; 2) represents the characteristics so that they perceptually coalesce into a single unit
  - Principle "See the Whole"; 2 aspects: completeness and unification



E.g. a good representation: all characteristics are represented by a single line, which is perceived as a unit

But… such a representation is seldom achievable

# Data Complexities

- ☹Multi-dimensionality (more than one referrer)
- ☹Multiple attributes
- ☹Large data volume (number of references in the reference set)
- ☹Complex, heterogeneous nature of referrers (e.g. geographical space)
- ☹Outliers, discontinuities, …

# Example: Behaviour over a Two-Dimensional Reference Set

Referrers

Space (set of states of the USA)

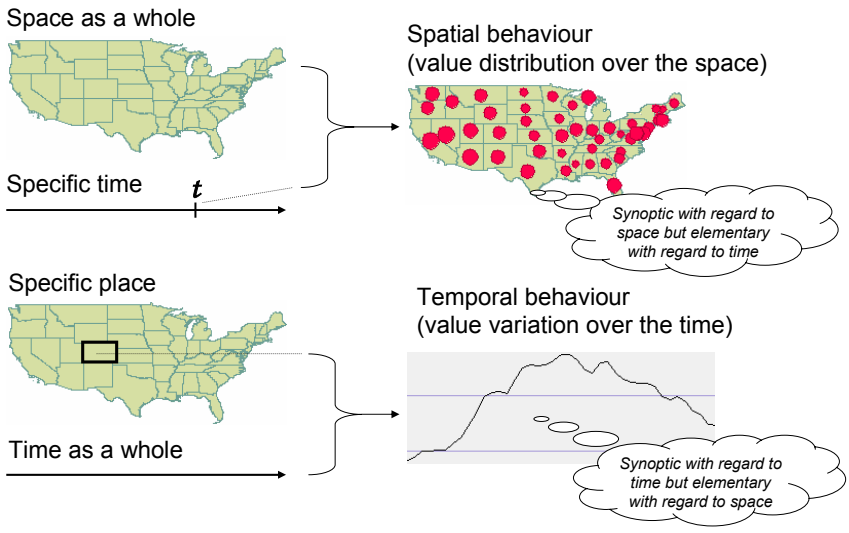
Time (set of years from 1960 to 2000)

Attributes

- Property crime rate
- Violent crime rate
- …

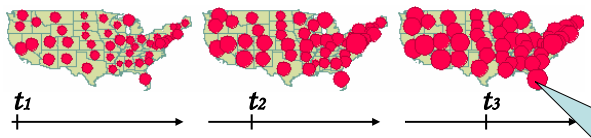
The behaviour cannot be represented as a single unit

# Slices of the Behaviour

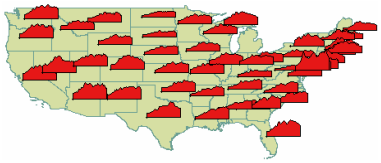



# Aspectual Behaviours

Aspect 1:
Temporal variation of the spatial behaviour



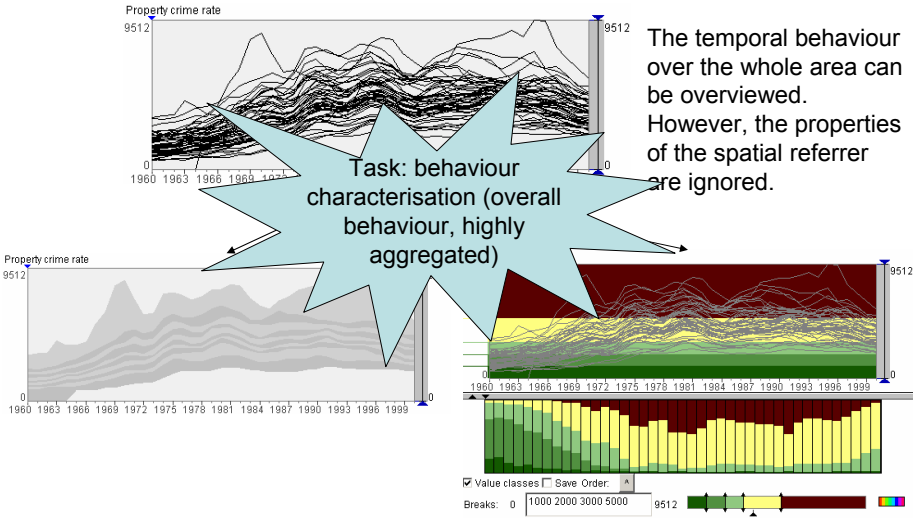

Aspect 2:
Spatial variation of the temporal behaviour



Tasks: behaviour characterisation (aspectual behaviours)

Completeness: both aspects must be characterised
Unification: not achieved

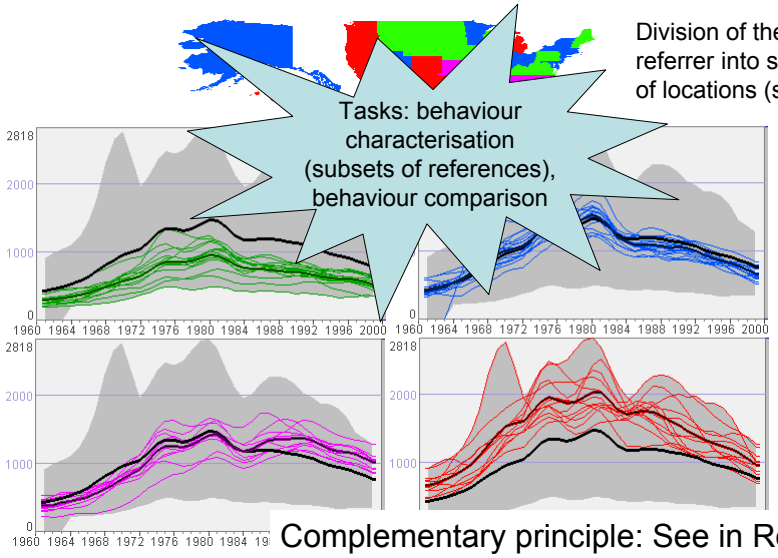
# Principle: Simplify and Abstract




Task: behaviour characterisation (overall behaviour, highly aggregated)

The temporal behaviour over the whole area can be overviewed. However, the properties of the spatial referrer are ignored.

# Principle: Divide and Group

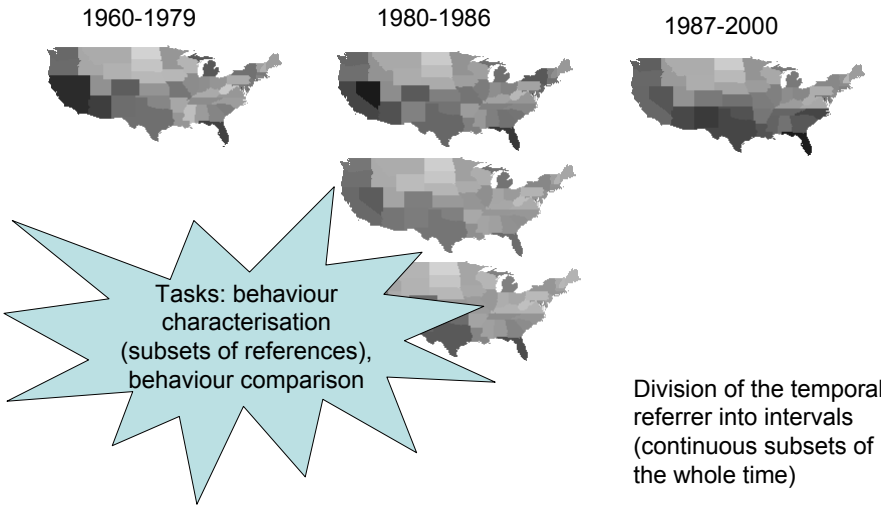



Tasks: behaviour characterisation (subsets of references), behaviour comparison

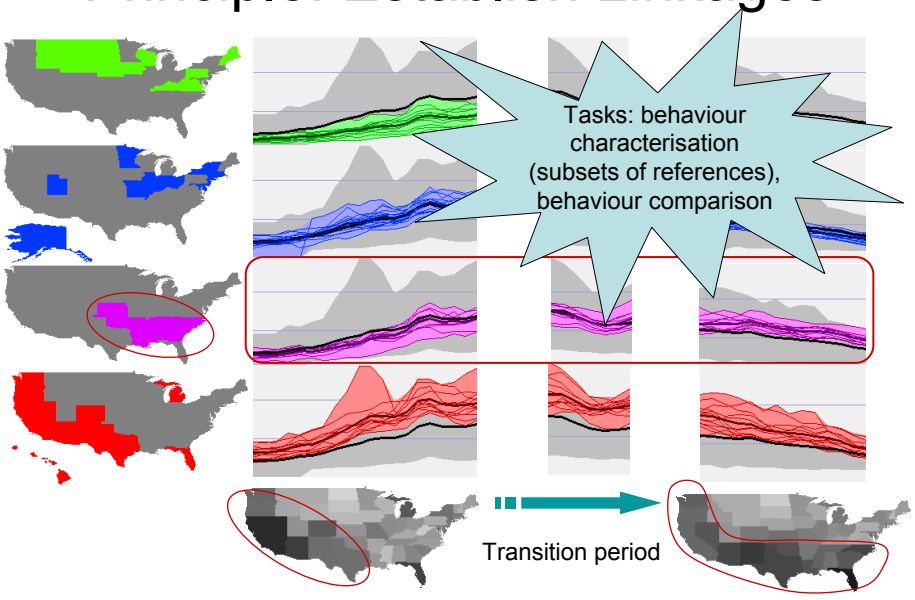
Division of the spatial referrer into subsets of locations (states)

Complementary principle: See in Relation

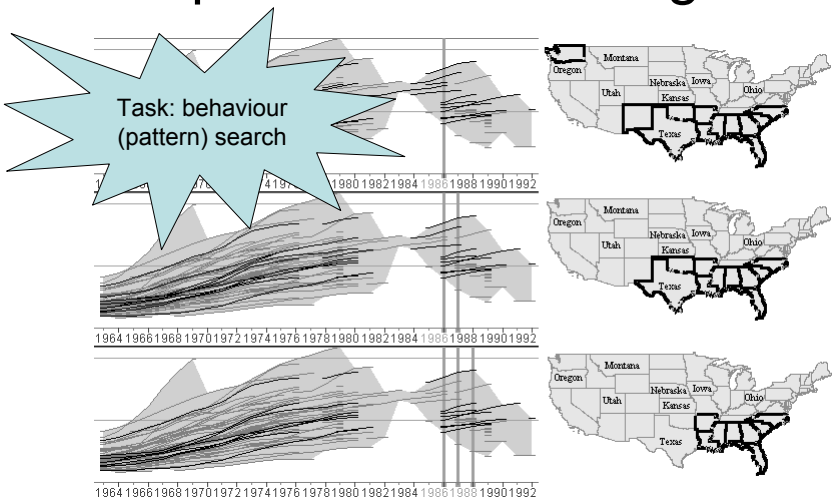
# Divide and Group (cont.)



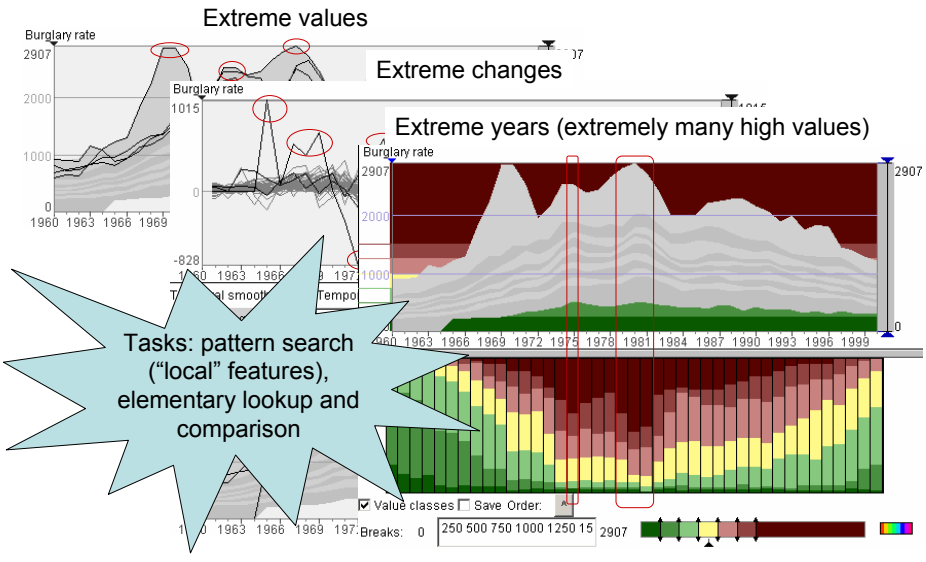

# Principle: Establish Linkages



# Principle: Look for Recognisable

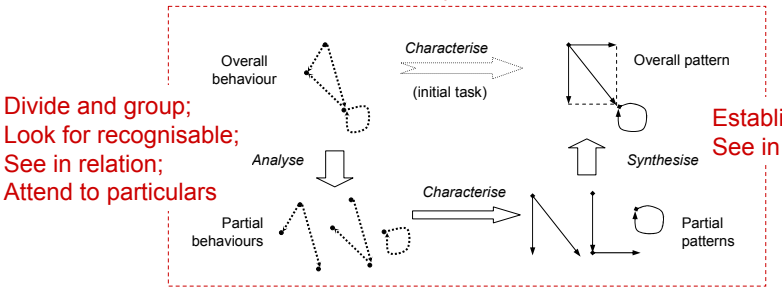



# Principle: Attend to Particulars



# EDA: Analysis and Synthesis

See the whole;
Simplify and abstract




Divide and group;
Look for recognisable;
See in relation;
Attend to particulars

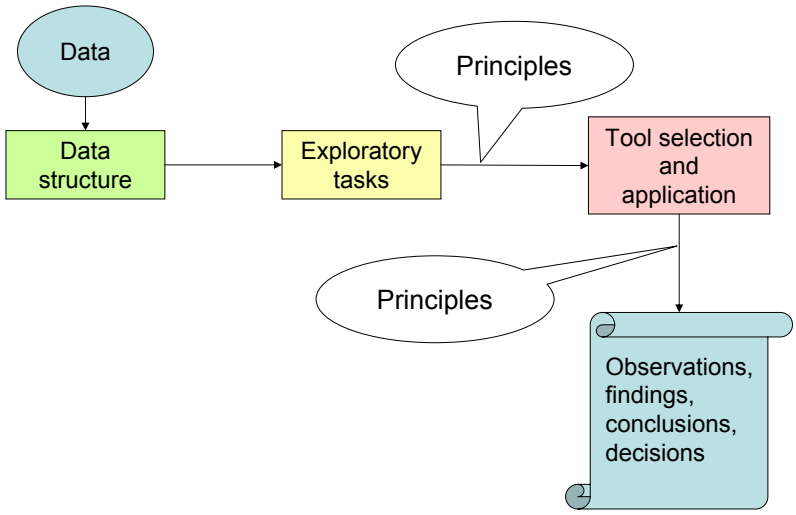
Establish linkages;
See in relation

Zoom and focus;
See in relation;
Attend to particulars

# Conclusion

- Dual use of the theory
  - Guidance for data analysts (tool users)
  - Guidance for tool designers
- Open issues
  - Human factors
  - Tool deficiencies

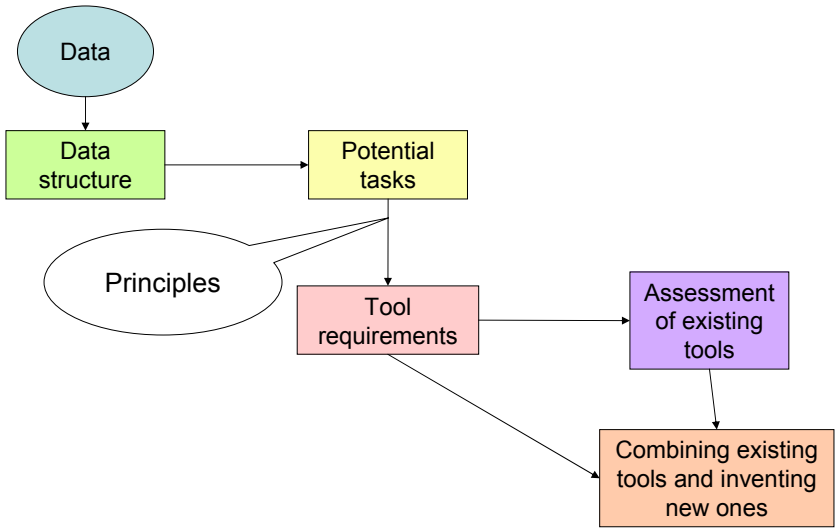
# Guidance for Data Analysts




# Open Issues (Human Factors)

- Lack of knowledge of the EDA concept
- Unconventional tools and approaches
- Complexity of the EDA process: many tasks, complex data ⇒ many different tools ⇒ difficult to master, to choose, and to combine
- Primacy of graphical techniques ⇒ main results are perceptual impressions ⇒ hard to capture, represent, and communicate
  - How to report about the work done?
- The results have the flavour of subjectivity and do not produce a solid impression (unlike e.g. results from using statistical methods)
  - "Serious" analysts are reluctant to use the EDA techniques
- Inexperienced users may jump to conclusions on the basis of just a single (default) visualisation instead of performing systematic, comprehensive EDA

# Guidance for tool designers




# General Requirements to EDA Software

- Space- and time-awareness
- Work with multidimensional data
- Work with uncertain and incomplete data
- Scalability
- Support and encouraging of multiple complementary views
- Easy tool linking and coordination
- Support of different levels of analysis, from “see the whole” to “attend to particulars”
- Support of the whole chain: exploration and hypothesis generation, computational analysis and hypothesis testing, presentation of findings and conclusions

# Open Issues (Tools)

- Work with qualitative (non-numeric) data
- Work with fuzzy, uncertain, and incomplete data
- Continue scalability efforts
- Embedded intelligence:
  - Know the principles and prompt the users to fulfil them
  - Know the tools and assist the users in choosing and utilising them
  - Relieve the users from the cognitive complexity of the EDA process
  - Adapt to user, data, tasks, and hardware
- Support in the capture and management of observations: recording, structuring, browsing, searching, checking, linking, interpreting…
- Link to confirmatory methods (hypotheses testing)
- Help in presentation and communication of observations, discoveries, conclusions, and decisions